IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Avadhanam et al.	Examiner:	Srirama T. Channavajjala
Serial No.:	10/830,164	Group Art Unit:	2166
		Confirmation No.:	8149
Filed:	April 21, 2004	Docket No.:	MS167378.02/40062.128USC1
Title:	METHOD AND SYSTEM FOR CREATING A DATABASE TABLE INDEX USING MULTIPLE PROCESSING UNITS		

CERTIFICATE UNDER 37 CFR 1.8:

I hereby certify that this correspondence is being transmitted via EFS-Web to the U.S. Patent Office on November 28, 2007.

By: Krystle Henley
Name: Krystle Henley

PRELIMINARY AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

In response to the Office Action mailed June 28, 2007, please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims that begins on page 2 of this paper.

Remarks begin on page 9 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

Listing of Claims:

1. (Currently amended) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein each processing unit has access to the table, the method comprising:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation;

accessing the table records in parallel, wherein each processing unit accesses each of the records;

filtering the accessed records in parallel, wherein each processing unit determines which records to keep;

independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units;

merging the sub-indexes together to create a final index related to the table; and

storing the final index for later use in locating records.

2. (Original) A method as defined in claim 1 wherein the act of creating the sub-indexes further comprises sorting the records and generating a data structure based on the sorted records.

3. (Original) A method as defined in claim 2 wherein the data structure is a B-Tree data structure.

4. (Original) A method a defined in claim 2 wherein the data structure has multiple levels.

5. (Original) A method as defined in claim 2 wherein the data structure is a clustered index.

6. (Original) A method as defined in claim 1 further comprising gathering sub-index statistical information and stitching sub-index statistical information.

7. (Original) A method as defined in claim 1 wherein the method is initiated by and index creation manager module.

8. (Original) A method as defined in claim 1 wherein the method is initiated by a query manager in response to a supplied query.

9. (Original) A method as defined in claim 1 wherein the method initiated automatically in response to a modification to the table.

10. (Currently Amended) A method as defined in claim 1 wherein the act of determining partition delimiters comprises:

sampling the table records to determine an approximate distribution of the values in [[the]] a key field;

creating a histogram based on the sampled information; and

evaluation the histogram to determine the partition delimiters.

11. (Original) A method as defined in claim 10 further comprising:

determining a processing unit goal value based on the number of processing units in the computer system;

determining a least common multiple value based on the processing unit goal value;

determining whether the histogram information may be substantially evenly split into the least common multiple value number of partitions;

if so, creating the partition delimiters based on the least common multiple value; and

if not, adjusting the processing unit goal to determine a new least common multiple value to determine partition delimiters.

12. (Currently Amended) A computer [[readable]] storage medium readable by a computer and encoding instructions for executing the method recited in claim 1.

13. (Currently Amended) A computer [[readable]] storage medium readable by a computer and encoding instructions for executing the method recited in claim 11.

14. (Currently Amended) A system for database table index creation for a database table, the database table stored in memory and comprising a plurality of records, the system comprising:

a plurality of processing units that respectively accesses the database table in parallel, the respective processing units accesses each of the records and filters the accessed records to determine which records to keep and wherein each of the respective processing units creates a sub-index of database table records resulting in a plurality of sub-indexes;

a merge tool that merges the plurality of sub-indexes into a final database table index; and

a store tool that stores the final database table index for later use in locating records.

15. (Original) A system as defined in claim 14 wherein each processing unit further comprises:

a scanning module that scans the database table;

a filter module that filters the accessed records and selectively predetermined records; and

a sorting module that sorts records kept by the filter module into a sub-index.

16. (Original) A system as defined in claim 15 wherein the scanning module, filter module and sorting module, for each processing unit, operates concurrently.

17. (Original) A system as defined in claim 15 further comprising a sampling module for sampling the database table and a partition module for dividing the records into substantially equal quantities related to the number of processing units.

18. (Currently Amended) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein more than one processing unit has access to the table, the method comprising:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, wherein at least one partition is dedicated to a first processing unit for index creation and at least one other partition is dedicated a second processing unit for index creation;

the first processing unit accessing a table record and determining whether the table record is associated with the at least one partition dedicated to the first processing unit;

the first processing unit only processing the accessed table record when the accessed table record is associated with the at least one partition dedicated to the first processing unit; and

storing a result produced by the first processing unit <u>for later use in locating records</u>.

19. (Original) A method as defined in claim 18 further comprises:

upon determining that the accessed table record is not associated with the at least one partition dedicated to the first processing unit, passing the accessed record to the second processing unit for index creation.

20. (Currently amended) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein more than one processing unit has access to the table, the method comprising:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation;

independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units;

allocating blocks of a disk to store each sub-index, wherein parts of each sub-index are stored on consecutive blocks on the disk;

merging the sub-indexes together to create a final index related to the table; and

storing the final index for later use in locating records.

21. (Original) A method as defined in claim 20 wherein the act of allocating portions of the disk allocates a predetermined number of blocks, the predetermined number of blocks is determined during the determination of the partition delimiters.

22. (Original) A method as defined in claim 20 wherein the allocation of portions of the disk comprises:

maintaining a cache of allocated pages and allocating pages for each partition in the cache for each processing unit; and

retrieving a pre-determined number of database pages upon request, and wherein the number of pages to allocate upon each request is determined by the size of the cache.

23. (Original) A method as defined in claim 22 wherein the cache has a size depending on the size of the index being built and the number of currently available free pages in the system.

24. (Currently amended) In a computer system having a plurality of processing units, an index creation system for creating an index of information for a table of data records, the system comprising:

a sampling module that samples the table of data records to determine sub-index delimiters;

two or more index creation modules, each index creation module associated with a processing unit, each index creation module creates a sub-index resulting in a plurality of sub-indexes;

a merge module that merges the sub-indexes into a final index,

wherein each index creation module comprises:

an access module that accesses data records from the table of data records;

a filter module that filters data records according the sub-index delimiters to keep only relevant data records; and

a sorting module that sorts the relevant data records into a sub-index; and

a store module that stores the final index for later use in locating records.

25. (Original) A system as defined claim 24 further comprising a memory allocation module that allocates parts of memory for storing the sub-indexes, and wherein the memory allocation module allocates a predetermined number of parts, the predetermined number of parts is determined during the determination of the delimiters.

26. (Original) A system as defined in claim 24 further comprising a cache memory module that manages a cache of allocation pages and allocates pages for storing each sub-index

in the cache and wherein the number of pages allocated to the cached is determined upon determining the delimiters.

27. (Currently Amended) An index creation system for creating an index of information for a table of data records, the system comprising:

means for sampling the table of data to determine sub-index delimiters;

means for accessing data records from the table in parallel;

means for filtering accessed data records to keep only relevant records;

means for creating two or more sub-indexes of relevant records;

means for merging the sub-indexes together; and

means for storing the merged sub-indexes for later use in locating records.

28. (Original) An index creation system as defined in claim 27 further comprising:

means for allocating memory for storing parts of each sub-index in contiguous memory blocks.

REMARKS

This Amendment and the following remarks are intended to fully respond to the Office Action mailed June 28, 2007, hereinafter "Office Action." In that Office Action, claims 1-28 were examined and all claims were rejected. More specifically, claims 1-28 were rejected under 35 U.S.C. § 101 as being non-statutory subject matter; and claims 1-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Gupta et al., US Patent No. 6,438,562 filed on August 24, 1999,hereinafter "Gupta," in view of Blank et al., US Patent No. 5,842,208 published on November 24, 1998, hereinafter "Blank."

Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, claims 1, 10, 12-14, 18, 20, 24, and 27 have been amended and no claims have been added or canceled. Therefore, claims 1-28 remain present for examination.

Summary of Interview

Applicants would like to thank the Examiner for his comments during and Examiner initiated phone conversation on or about April 12, 2007. Examiner made suggestions regarding clarifying certain claim language. No agreement was reached.

Interview Request

In order to further prosecution, applicants have submitted an in-person interview request with this amendment. The Applicant's representatives will be in Washington D.C. on December 14th. If that date does not work for the Examiner, Applicants request a telephonic interview.

The interview request has been faxed to (571) 273-8300, the fax number for the organization where the application or proceeding is assigned.

Claim Rejections – 35 U.S.C. § 101

Claims 1-28 were rejected under 35 U.S.C. § 101 because the Office Action asserts that they are directed to non-statutory subject matter. Claims 1 and 20 have been amended to recite

storing the final index for later use in locating records; claim 14 has been amended to recite a store tool that stores the final database table index for later use in locating records; claim 18 has been amended to recite storing a result produced by the first processing unit for later use in locating records; claim 24 has been amended to recite a store module that stores the final index for later use in locating records; and claim 27 has been amended to recite means for storing the merged sub-indexes for later use in locating records. The claims recite useful, concrete, and tangible results. *State Street Bank and Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998). Namely, the indexes allow queries to be resolved much more quickly by providing relatively short paths to desired information. (Application, p.2, ll. 3-5). Storing the index allows another query to make use of the index to access information without having to first reconstruct the index. This saves additional time and allows an even quicker resolution of future queries.

Claims 12-13 have been amended to recite a computer storage medium. The specification states,

> Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Memory 24, removable storage 28 and non-removable storage 30 are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by system 20. (Specification, p. 8, ll. 5-13).

Thus, the embodiments recited in the amended claims are encoded in tangible, computer-readable media and are, thus, patentable subject matter under 35 U.S.C. § 101. *In re Beauregard*, 53 F.3d 1583, 1584 (Fed. Cir. 1995). Applicant's respectfully request that the Examiner withdraw the § 101 rejection and issue a notice of allowance, for all claims.

Claim Rejections – 35 U.S.C. § 103(a)

Claims 1-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Gupta in view of Blank. Applicants respectfully traverse the § 103(a) rejections because either the

Examiner failed to state a *prima facie* case of obviousness or the current amendments to the claims now render the Examiner's arguments moot. To establish a *prima facie* case of obviousness under 35 U.S.C. § 103(a), the references must teach or suggest all of the claimed limitations to one of ordinary skill in the art at the time the invention was made. M.P.E.P §§ 2142, 2143.03; *In re Royka*, 490 F.2d 981, 985 (C.C.P.A. 1974); *In re Wilson*, 424 F.2d 1382, 1385 (C.C.P.A. 1970). Further, under *KSR Int'l Co. v. Teleflex, Inc.*, there "must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." 127 S. Ct. 1727, 1741 (2007). Neither Gupta nor Blank, either separately or in combination, teach or suggest all of the limitations of the recited claims.

Gupta relates to "a method, system, and product for coordinating parallel update for a global index of and indexed table." (Gupta, Abstract). "Techniques for maintaining a *global index* of a table during parallel data manipulations operations involve a coordinator process, data manipulation slaves and index update slaves. The coordinator process distributes data manipulation operations among a plurality of data manipulation slaves." (Gupta, col. 8, ll. 1-6) (emphasis added). Maintenance of the global index is performed with the use of Data Manipulation Language ("DML") commands. (*See* Gupta, col. 4, ll. 43-45). Gupta specifically states that the DML commands supported are "commands to delete rows, insert rows, and update rows." (Gupta, col. 4, ll. 47-48). No mention is made of creating actual indexes. Instead, Gupta teaches a method using parallel DML ("PDML") operations that accomplishes the "need to update a global index as a result of PDML operations without suffering the deficiencies of lost clustering, or contention for the same block, the latter leading to excessive waits or block pinging." (*See* Gupta, col. 7, ll. 35-38).

Gupta does not teach a parallel *processing unit* environment. Instead, Gupta teaches parallel *processes* that are executed by a single processing unit. (*See* Gupta, FIG. 4). These multiple processes are executed by the single processing unit. (*See* Gupta, col. 9, l. 23 – col. 10, l. 16). Gupta also teaches that indexes may be stored as B-Trees. (Gupta, FIG. 3).

Blank relates to a "recover/build index system [that] builds an index for a file by scanning partitions of the file in parallel to retrieve key/rid values. The recover/build index system then sorts the scanned key/rid values for each partition in parallel." (col. 1, ll. 37-41).

After the data is sorted in parallel, a "merge program merges the sort streams received from the sort programs to create a merge stream. The merge program accepts the sort stream from two or more sort programs. The merge program then passes the merge stream to an index build program." (col. 3, ll. 10-14). Thus, Blank teaches a method where a parallel sort is merged via combining the data streams produced by two or more sorts into a single data stream. Blank then performs *index creation on this single data stream.*

Claim 1

Independent claim 1 is allowable over the cited art because the cited art fails to teach or suggest all of its limitations. For example, the cited references fail to teach or suggest at least determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation. Gupta cannot teach this limitation because the reference only teaches the use of one processing unit. The Office Action points to a portion of Gupta related to multiple slave processes to show this limitation. (col. 14, ll. 44-50 and 54-56). A process is not a processing unit. Blank fails to compensate for this deficiency. Claim 1 provides that the table is separated into non-overlapping partitions of records. In part, the fact that the partitions do not overlap permits the method of claim 1 to properly create multiple sub-indexes. On the other hand, because Blank does not create sub-indexes for each partition it is not necessary to ensure that the partitions do not overlap. Blank appears to be performing a "divide-and-conquer" sort algorithm using multiple processing units where it is unnecessary to ensure that the data partitions do not overlap.

The references also fail to teach or suggest at least accessing the table records in parallel, wherein each processing unit accesses each of the records. Clearly, Gupta cannot teach or suggest this limitation because the reference only teaches the use of one processing unit. However, the Office Action cites Gupta as teaching this limitation. The portion of Gupta cited by the Office Action as teaching this limitation relates to a single coordinator process that distributes data manipulation tasks among a number of data manipulation slaves and index update slaves. (*See* Gupta, col. 8, ll. 1-13). The *single* coordinator process directing multiple slaves is not the same as each processing unit accesses each of the records. First, there is no

suggestion in Gupta that each of the slaves accesses each record in the table. Second, even if each record was accessed by each slave process in Gupta (which does not appear to be true), Gupta is teaching *multiple processes, not processing units*.

Although not cited by the Office Action, Blank does not compensate for this deficiency. While multiple processing units are taught in Blank, the reference also teaches that each processing unit accesses only a portion of the table, i.e., each processing unit scans a single partition. Blank teaches,

> [t]he scan programs **108** executing in parallel extract key values (of a particular key) and record identifiers (rids) or pointers from the partitions **120** to create a key/rid or scan stream *for each partition* **112**. (Blank, col. 2, l. 64 – col. 3, l.1) (emphasis added).

The scan programs in Blank are only assigned a particular partition of the table, not each of the records in the table.

The references also fail to teach or suggest at least filtering the accessed records in parallel, wherein each processing unit determines which records to keep. Again, Gupta cannot possibly teach this because the reference teaches the use of a single processing unit. However, the Office Action again cites Gupta as teaching this limitation. The portions of Gupta cited by the Office Action as teaching this limitation instead relate to a single coordinator process determining a range of key values and assigning slaves to each range. (Gupta, col. 7, ll. 45-51). Again, the coordinator process and the slaves are under the control of a single processing unit making it impossible for Gupta to teach this limitation. Furthermore, Blank cannot compensate for this deficiency because it assigns processing units to different partitions. Because each processing unit does not scan each record of the table, Blank does not teach or suggest each processing unit determines which records to keep.

Additionally, the references do not teach or suggest at least merging the sub-indexes together to create a final index related to the table. In fact, the Office Action assents that Gupta does not teach this. (*See* Office Action, p. 9). Instead, the Office Action relies upon Blank as teaching this limitation. Blank does not teach the creation of sub-indexes, thus making it

impossible for the reference to teach or suggest merging the sub-indexes together to create a final index related to the table. In fact, Blank does the opposite. Before creating an index, Blank scans and sorts multiple partitions of a database. These resulting sorted data streams are then merged together into a single data stream. Blank teaches creating the index based off of the *single data stream* that results from merging the various sort streams together. The sort streams are not indexed, and therefore cannot be considered sub-indexes.

The Office Action argues that both Gupta and Blank teach sub-indexes. Applicants respectfully disagree. The Office Action asserts that the B-tree taught in Gupta provides an example of sub-indexes. (*See* Office Action, pp. 22-23). The Office Action is apparently equating each node of a B-Tree to a separate sub-index. Applicants disagree. First, the B-tree is used to store a *final index*. Second, the B-tree is simply a data structure used for storing the index. B-trees are data structures commonly used to store database information. This is due to the ability of B-trees to become unbalanced, thus allowing for a logarithmic lookup time when accessing the B-tree. To maintain a balanced structure, B-trees may add empty data or reorganize data, thus leaving the possibility of empty nodes in the B-tree.

Although the data structure is divided into nodes, these nodes are all a part of the single index. B-trees, like all trees, have a hierarchical structure. This hierarchical structure is inherent to the structure, not a result of multiple sub-indexes. The Office Action is making a logical leap in declaring that the nodes of the storage structure are separate sub-indexes. Databases are generally large and span multiple storage devices. One would not say that a database spanning multiple storage devices is actually multiple databases wherein each database corresponds to a separate storage device. In this same line of logic, a single index divided due to the properties of the data structure it is stored upon cannot be said to be multiple sub-indexes.

Applicants also respectfully disagree with the Office Action's assertion that Blank teaches sub-indexes. (*See* Office Action, p. 24). As previously mentioned, Blank teaches separate sort streams that are combined into a single data stream. This single data stream, not the sort streams, is indexed. The Office Action appears to equate the sort streams to sub-indexes. However, the sort streams are clearly not sub-indexes because they are not indexed. In fact, Blank teaches away from the creation of multiple sub-indexes, instead teaching merging sorted

stream *before* creating an index of the data. The claimed subject matter uses multiple processing units to create multiple indexes, which is an efficiency benefit lost in Blank due to its indexing of a single data stream.

For the above mentioned reasons, Applicants maintain that the cited references also fail to teach or suggest at least independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units. For at least these foregoing reasons, independent claim 1 is allowable over the cited references.

Claim 14

Independent claim 14 is allowable over the cited references. Claim 14 recites, *inter alia*, a plurality of processing units that respectively accesses the database table in parallel, the respective processing units accesses each of the records and filters the accessed records to determine which records to keep and wherein each of the respective processing units creates a sub-index of database table records resulting in a plurality of sub-indexes. As previously mentioned, Gupta cannot teach or suggest this limitation because the reference only teaches the use of a single processing unit. Although multiple processes are taught, these processes all rely on a single processing unit. Thus, Gupta does not provide the processing efficiency of the claimed limitation. Additionally, because each processing unit in Blank only scans a partition of a table, Blank cannot teach or suggest the respective processing units accesses each of the records and filters the accessed records to determine which records to keep and wherein each of the respective processing units creates a sub-index of database table records resulting in a plurality of sub-indexes.

Again, because neither reference teaches sub-indexes, the references also fail to teach or suggest a merge tool that merges the plurality of sub-indexes into a final database table index. The references, nether alone nor in combination, teach or suggest at least these limitations of claim 14. For at least these reasons, claim 14 is allowable over the cited references.

Claim 18

Independent claim 18 is allowable over the cited references. For example, the references fail to teach or suggest at least determining partition delimiters, each partition delimiter

separating the table into non-overlapping partitions of records, wherein at least one partition is dedicated to a first processing unit for index creation and at least one other partition is dedicated a second processing unit for index creation. As previously mentioned, Gupta cannot teach or suggest this limitation because the reference teaches the use of a single processing unit. Furthermore, Blank cannot compensate for this deficiency. Blank teaches sorting multiple partitions. However, before creating an index, these partitions are merged together. Thus, Blank teaches creating an index for the *entire table*, not creating indexes for *partitions of the table*. Thus, the references, either alone or in combination, fail to teach or suggest all of the limitations of claim 18. For at least this reason, claim 18 is allowable over the cited references.

Claim 20

Independent claim 20 is allowable over the cited references. The references fail to teach or suggest at least determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation. Again, Gupta cannot teach this because the reference teaches a single processing unit. Blank cannot compensate for this deficiency because it does not teach or suggest non-overlapping partitions. Because Blank only scans and sorts the partitions, it is not necessary that the partitions are non-overlapping. On the other hand, because embodiments of the present disclosure create sub-indexes for each partition, it is necessary to ensure that each partition contains non-overlapping data.

Additionally, the references fail to teach or suggest at least independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units. As previously noted, Applicants maintain that neither of the cited reference teach or suggest the creation of sub-indexes. The nodes of a B-tree used to store indexes in Gupta do not constitute separate sub-indexes and Blank teaches merging the sorted partitions together before creating a final index. In light of this, the references also cannot teach or suggest, either alone or in combination, merging the sub-indexes together to create a final index related to the table. For at least the forgoing reasons, claim 20 is allowable over the cited references.

Claim 24

Independent claim 24 is allowable over the cited references. The cited references do not teach or suggest, either alone or in combination at least, two or more index creation modules, each index creation module associated with a processing unit, each index creation module creates a sub-index resulting in a plurality of sub-indexes. Gupta cannot teach this limitation because it only teaches a single limitation. Blank cannot compensate for this deficiency because, as previously noted, the reference teaches merging different sort streams into a single stream before creating a final index, not creating multiple sub-indexes and merging the sub-indexes into a final index. For at least these reasons, the references also do not teach or suggest a merge module that merges the sub-indexes into a final index. For at least these reasons, claim 27 is allowable over the cited references.

Claim 27

Independent claim 27 is allowable over the cited references. Claim 27 recites means for creating two or more sub-indexes of relevant records; and means for merging the sub-indexes. For the previously stated reasons, Applicants maintain that neither reference teaches or suggests sub-indexes. Furthermore, Gupta does not teach means for merging the sub-indexes. Blank does not compensate for this deficiency because Blank teaches merging sort streams, not sub-indexes, and the creating an index from the merged sort streams. For at least the forgoing reasons, claim 27 is allowable over the cited references.

For the forgoing reasons, neither Gupta nor Blank, either alone or in combination, teach all of the limitations of independent claims 1, 14, 18, 20, 24, and 27 or therefore cannot anticipate the present invention as claimed. Claims 1, 14, 18, 20, 24, and 27 are allowable over the prior art of record and should be allowed. All other claims, *i.e.*, claims 2-13, 15-17, 19, 21-23, 25-26, and 28 depend from one of the allowable independent claims and are, thus, also allowable over the prior art of record. Therefore, Applicants respectfully request that the Examiner issue a notice of allowance, for all claims, at his earliest convenience.

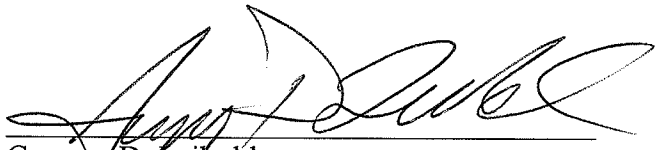
Conclusion

This Amendment fully responds to the Office Action mailed on June 28, 2007. Still, that Office Action may contain arguments and rejections that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, failure of this Amendment to directly address an argument raised in the Office Action should not be taken as an indication that the Applicants believe the argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is respectfully requested to telephone the undersigned to attempt to resolve those issues.

Respectfully submitted,



Date: November 28, 2007

Gregory D. Leibold
Reg. No. 36,408
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
(303) 357-1642

NOV 2 8 2007

Merchant & Gould

An Intellectual Property Law Firm

Merchant & Gould P.C.
1950 Independence Plaza
1050 Seventeenth Street
Denver, Colorado 80265-0100

TEL 303.357.1642
FAX 303.357.1671
www.merchantgould.com

Fax Transmission | November 28, 2007

To:	Examiner Channavajjala	**From:**	Gregory D. Leibold
Company:	USPTO	Our Ref.:	40062.0128USC1
Your Ref:	10/830,164	Fax No.:	303.357.1671
Fax No.:	571.273.8300	Phone No.:	303.357.1642
Phone No.:	571.272.4108	Total Pages:	2
State/Country:	USA	E-Mail:	gleibold@merchantgould.com
Confirmation Via Mail:	☐ Yes ☑ No	Return Fax To:	Gregory D. Leibold

Document Transmitted: Applicant Initiated Interview Request Form

NOV 2 8 2007

PTOL-413A (09-06)
Approved for use through 03/31/2007 OMB 0651-0031
U.S. Patent and Trademark Office U.S. DEPARTMENT OF COMMERCE

Applicant Initiated Interview Request Form

Application No.: 10/830,164 First Named Applicant: Avadhanam, et al.
Examiner: Channavajjala Art Unit: 2166 Status of Application: Pending

Tentative Participants:
(1) Examiner Channavajjala (2) Gregory Leibold
(3) Andrew Pouzeshi (4)

Proposed Date of Interview: 12/14/2007 **Proposed Time:** 10:00 (AM/PM)

Type of Interview Requested:
(1) [ ] Telephonic (2) [X] Personal (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [ ] YES [X] NO
If yes, provide brief description:

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) §103 Rej.	1-28	Gupta and Blank	[ ]	[ ]	[ ]
(2)			[ ]	[ ]	[ ]
(3)			[ ]	[ ]	[ ]
(4)			[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:
In the interest of furthering prosecution, Applications suggest discussing the differences between Gupta and Blank and the claims of the present application.

An interview was conducted on the above-identified application on or about April 12, 2007.
NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).
This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

Applicant/Applicant's Representative Signature

Examiner/SPE Signature

Gregory D. Leibold
Typed/Printed Name of Applicant or Representative

36,408
Registration Number, if applicable